

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the instant application:

Listing of Claims:

1. (Currently Amended) A method of scheduling pattern-based ~~a request for a plurality of~~ Web services comprising the steps of:

receiving a request for a plurality of Web services;

extracting a pattern object from the request and placing the extracted pattern object in a work area, the pattern object specifying a plurality of Web services to be invoked;

provisioning one or more termination rules within a termination watcher, wherein the termination watcher removes the pattern object from the work area if it is detected that the pattern object conforms to one or more of the termination rules;

(a) providing a plurality of service activation rules accessible by a scheduler, each service activation rule specifying a trigger condition and a state condition for causing a watcher to ~~invoke~~ and a particular corresponding Web service to be activated;

(b) receiving in the scheduler at least one event generated dynamically and indicating a change in the work area ~~a common memory~~, wherein each event specifies trigger information;

(c) ~~comparing the trigger conditions of the service activation rules with the trigger information of the at least one event;~~

(d) ~~adding service activation rules that match the at least one event to a trigger list;~~

(e) identifying service activation rules having trigger conditions and state conditions that match the received event and corresponding pattern object; comparing

~~the state conditions of service activation rules in the trigger list with a state of the common memory; and~~

(f) causing watchers specified by the identified service activation rules to execute and thus invoke corresponding Web services; and

running the Web services until a termination criterion is detected by the termination watcher ~~selecting the service activation rules of the trigger list that match the state of the common memory, thereby causing the invocation of at least one watcher and corresponding Web service, wherein the state of the common memory is dictated by at least one pattern object.~~

2. (Currently Amended) The method of claim 1, wherein each event indicating a change in the work area ~~common memory~~ is associated with the pattern object.

3. (Cancelled).

4. (Currently Amended) The method of claim 1, further comprising the steps of:
at least one of the watchers modifying the pattern object in the work area ~~common memory; and~~

the work area ~~common memory~~ sending at least one event indicating a state change, wherein each event specifies a trigger condition; ~~and~~
~~repeating said steps (b)-(f).~~

5. (Currently Amended) The method of claim 4, wherein the at least one of the watchers modifies the pattern object in the work area ~~common memory~~ according to instructions from an associated one of the Web services.

6. (Cancelled).

7. (Original) The method of claim 1, wherein each pattern object specifies at least two Web services to be performed.

8. (Currently Amended) The method of claim ~~[[1]]~~ 7, wherein the at least two ~~watchers each invoke an associated~~ Web services ~~service~~ to operate concurrently with one another ~~in said step (f)~~.

9. (Currently Amended) The method of claim ~~[[1]]~~ 7, wherein the at least two ~~watchers each invoke an associated~~ Web services ~~service~~ to operate sequentially ~~in said step (f)~~.

10. (Cancelled).

11. (Currently Amended) The method of claim 1, wherein ~~further comprising the step of~~ at least one of the watchers modifies ~~modifying~~ the pattern object according to instructions from an associated one of the Web services.

12. (Currently Amended) The method of claim 1, ~~wherein said step (f) includes matched service activation rules within an execution list, said method~~ further comprising the steps of:

receiving an event list of one or more events from the work area;
identifying trigger information from events in the event list;
comparing the trigger information with trigger conditions specified by service activation rules;

adding matching service activation rules to a trigger list;
identifying state information from the pattern object in the work area;
comparing the state information of the pattern object with state conditions of the
service activation rules that have been added to the trigger list;
adding matching service activation rules to an execution list; and
selecting watchers specified by service activation rules in the execution list for
execution, causing each watcher to invoke a corresponding Web service
~~identifying service activation rules in the execution list corresponding to~~
~~competitive Web services;~~
~~comparing the identified service activation rules with at least one service selection~~
~~rule, wherein the at least one selection rule comprises a heuristic evaluation of the~~
~~competitive Web services; and~~
~~invoking watchers specified by the identified service activation rules according to~~
~~said comparing step.~~

13. (Currently Amended) [[A]] The method of Claim 12, further resolving conflicts
between competing Web services comprising the steps of:

scanning the execution list to determine whether any service activation rules for
competing Web services exist;

if no service activation rules for competing Web services exist, invoking all
watchers in the execution list;

if service activation rules for competing Web services exist, performing the
following steps:

~~reading an execution list of service activation rules corresponding to watchers,~~
~~wherein each watcher is configured to invoke an associated Web service;~~

identifying service activation rules in the execution list corresponding to ~~competitive~~ competing Web services;

~~comparing~~ selecting certain service activation rules from the identified service activation rules corresponding to competing Web services ~~with using~~ at least one service selection rule, wherein the at least one selection rule comprises a heuristic evaluation of the ~~competitive~~ competing Web services; and

invoking watchers specified by the ~~identified~~ selected service activation rules ~~and according to said comparing step.~~

14. (Currently Amended) A system for ~~processing complex requests for~~ scheduling pattern-based Web services comprising:

a plurality of service activation rules, each service activation rule specifying a trigger condition and a state condition for causing a watcher to invoke a particular Web service;

a server configured to receive a request for more than one Web ~~service~~ services;

at least one servlet configured to extract a pattern object from the request and to format a response to the request;

a common memory as a work area that temporarily stores the pattern object while the Web services specified by the pattern object execute, wherein said ~~common memory~~ work area dynamically generates events when content of said ~~common memory~~ work area is changed;

a plurality of watchers, each watcher corresponding to a particular Web service;
and

a scheduler configured to receive events from the work area, compare ~~trigger conditions specified by~~ said service activation rules with trigger information of the events, identify service activation rules having trigger conditions and state conditions that

~~match the received event and corresponding pattern object compare state conditions of said service activation rules in the trigger list with a state of the common memory, and select at least one of said service activation rules causing the invocation of at least one of said watchers and corresponding Web services; and~~

~~a termination watcher configured to be provisioned with one or more termination rules and to remove the pattern object from the work area if it is detected that the pattern object conforms one or more of the termination rules.~~

15. (Cancelled).

16. (Original) The system of claim 14, wherein said watchers are further configured to modify the pattern object according to instructions provided from an associated one of the Web services.

17. (Cancelled).

18. (Original) The system of claim 17, wherein said execution evaluation processor invokes at least one of said watchers according to selected service activation rules.

19. (Currently Amended) [[A]] The system of Claim 14, wherein the scheduler for processing complex requests for Web services comprising:

~~a plurality of service activation rules, each service activation rule specifying a trigger condition and a state condition for causing a watcher to invoke a particular Web service;~~

~~a common memory that temporarily stores a pattern object while Web services specified by the pattern object execute, wherein said common memory generates events when content of said common memory is changed;~~

a trigger evaluation processor configured to compare the trigger conditions of the service activation rules with trigger information from at least one event, wherein said trigger evaluation processor adds service activation rules that match the at least one event to a trigger list; and

a state evaluation processor configured to compare the state conditions of service activation rules in the trigger list with [[a]] state information of the pattern object, wherein said state evaluation processor adds service activation rules that match the state information of the pattern object to an execution list ~~common memory and cause the service activation rules of the trigger list that match the pattern object to be selected, thereby causing the invocation of at least one watcher and corresponding Web service.~~

20. (Cancelled).

21. (Currently Amended) The system of claim [[20]] 19, further comprising an execution evaluation processor configured to select service activation rules associated with competitive Web services according to service selection rules, wherein the selection rules comprises a heuristic evaluation of the competitive Web services.

22. (Original) The system of claim 21, wherein said execution evaluation processor invokes at least one of said watchers according to selected service activation rules.

23. (Cancelled) .

24. (Cancelled).

25. (Cancelled) .

26. (Currently amended) A computer-readable storage, having stored thereon a computer program having a plurality of code sections executable by a computer for causing the computer to perform the steps of:

receiving a request for a plurality of Web services;

extracting a pattern object from the request and placing the extracted pattern object in a work area, the pattern object specifying a plurality of Web services to be invoked;

provisioning one or more termination rules within a termination watcher, wherein the termination watcher removes the pattern object from the work area if it is detected that the pattern object conforms to one or more of the termination rules;

(a) providing a plurality of service activation rules accessible by a scheduler, each service activation rule specifying a trigger condition and a state condition for causing a watcher to ~~invoke~~ and a particular corresponding Web service to be activated;

(b) receiving in the scheduler at least one event generated dynamically and indicating a change in the work area ~~a common memory~~, wherein each event specifies trigger information;

(c) ~~comparing the trigger conditions of the service activation rules with the~~ trigger information of the at least one event;

(d) ~~adding service activation rules that match the at least one event to a trigger~~ list;

(e) identifying service activation rules having trigger conditions and state conditions that match the received event and corresponding pattern object; ~~comparing~~

~~the state conditions of service activation rules in the trigger list with a state of the common memory; and~~

(f) causing watchers specified by the identified service activation rules to execute and thus invoke corresponding Web services; and

running the Web services until a termination criterion is detected by the termination watcher ~~selecting the service activation rules of the trigger list that match the state of the common memory, thereby causing the invocation of at least one watcher and corresponding Web service, wherein the state of the common memory is dictated by at least one pattern object.~~

27. (Currently Amended) The computer-readable storage of claim 26, wherein each event indicating a change in the ~~common memory~~ work area is associated with the pattern object.

28. (Cancelled).

29. (Currently Amended) The computer-readable storage of claim 26, further causing the machine to perform the steps of:

at least one of the watchers modifying the pattern object in the work area ~~common memory; and~~

the work area ~~common memory~~ sending at least one event indicating a state change, wherein each event specifies a trigger condition; ~~and~~

~~repeating said steps (b)-(f).~~

30. (Currently Amended) The computer-readable storage of claim 29, wherein the at least one of the watchers modifies the pattern object in the work area ~~common memory~~ according to instructions from an associated one of the Web services.

31. (Cancelled).

32. (Previously Presented) The computer-readable storage of claim 26, wherein each pattern object specifies at least two Web services to be performed.

33. (Currently Amended) The computer-readable storage of claim ~~[[26]]~~ 32, wherein ~~the~~ at least two ~~watchers each invoke an associated~~ Web services ~~service to~~ operate concurrently with one another ~~in said step (f)~~.

34. (Currently Amended) The computer-readable storage of claim ~~[[26]]~~ 32, wherein ~~the~~ at least two ~~watchers each invoke an associated~~ Web services ~~service to~~ operate sequentially ~~in said step (f)~~.

35. (Cancelled).

36. (Currently Amended) The computer-readable storage of claim 26, wherein ~~further causing the machine to perform the step of~~ at least one of the watchers ~~modifying~~ modifies the pattern object according to instructions from an associated one of the Web services.

37. (Currently Amended) The computer-readable storage of claim 26, wherein said step (f) includes ~~matched service activation rules within an execution list, said machine readable storage further causing the machine to perform the steps of:~~

receiving an event list of one or more events from the work area;

identifying trigger information from events in the event list;

comparing the trigger information with trigger conditions specified by service activation rules;

adding matching service activation rules to a trigger list;

identifying state information from the pattern object in the work area;

comparing the state information of the pattern object with state conditions of the service activation rules that have been added to the trigger list;

adding matching service activation rules to an execution list; and

selecting watchers specified by service activation rules in the execution list for execution, causing each watcher to invoke a corresponding Web service

~~identifying service activation rules in the execution list corresponding to competitive Web services;~~

~~comparing the identified service activation rules with at least one service selection rule, wherein the at least one selection rule comprises a heuristic evaluation of the competitive Web services; and~~

~~invoking watchers specified by the identified service activation rules according to said comparing step.~~

38. (Currently Amended) The computer-readable storage of Claim 26, having stored thereon a computer program having a plurality of code sections executable by a machine for further causing the machine to perform the steps of:

scanning the execution list to determine whether any service activation rules for competing Web services exist;

if no service activation rules for competing Web services exist, invoking all watchers in the execution list;

if service activation rules for competing Web services exist, performing the following steps:

~~reading an execution list of service activation rules corresponding to watchers, wherein each watcher is configured to invoke an associated Web service;~~

identifying service activation rules in the execution list corresponding to ~~competitive~~ competing Web services;

~~comparing~~ selecting certain service activation rules from the identified service activation rules corresponding to competing Web services with using at least one service selection rule, wherein the at least one selection rule comprises a heuristic evaluation of the ~~competitive~~ competing Web services; and

invoking watchers specified by the ~~identified~~ selected service activation rules and according to said ~~comparing~~ step.